Amendments to the Claims:

1. (Previously Presented) A method of sampling instructions executing in a multi-threaded processor comprising:

selecting an instruction for sampling;

storing sampling information relating to the instruction;

determining whether the sampling information includes an event of interest to a particular thread within which the instruction is executing; and

reporting the sampling information to the particular thread when the sampling information includes an event of interest.

2. (Previously Presented) The method of claim 1 further comprising providing a register with a bit vector representing a plurality of events of interest; and

wherein the determining whether the sampling information includes the event of interest further includes comparing the sampling information relating to the instruction to the bit vector.

3. (Previously Presented) The method of claim 2 wherein the comparing is via at least one of a mask operation or a more expressive operation.

4. (Original) The method of claim 1 wherein the selecting the instruction is without regard to a thread to which the instruction is bound.

5. (Original) The method of claim 1 further comprising identifying a thread to which the instruction is bound when the instruction is selected.

6. (Original) The method of claim 1 further comprising providing filtering criteria on a per-thread basis.

7. (Original) The method of claim 1 further comprising providing a single set of filtering criteria; and, scheduling sampling among a plurality of threads via software.

8. (Currently Amended) A method of sampling instructions executing in a multi-threaded processor comprising:

setting a candidate counter to a number;

selecting an instruction for sampling;

storing information relating to the instruction;

determining whether all events for the instruction have occurred;

decrementing the candidate counter when all events for the instruction have occurred and when the instruction corresponds to a desired sampled thread;

determining whether the candidate counter equals zero; and

reporting the instruction to a particular thread when the candidate counter equals zero.

9. (Original) The method of claim 8 wherein the information relating to the instruction represents an instruction history, and the instruction history includes information relating to at least one of an events value, a program counter value, a branch target address value, an effective memory address value, a latency value, a number in issue bundle value, a number in retire bundle value, a privilege value, a branch history value and a number in fetch bundle value.

10. (Original) The method of claim 8 wherein the selecting the instruction is without regard to a thread to which the instruction is bound.

11. (Original) The method of claim 8 further comprising identifying a thread to which the instruction is bound when the instruction is selected.

12. (Original) The method of claim 8 further comprising providing filtering criteria on a per-thread basis.

13. (Original) The method of claim 8 further comprising providing a single set of filtering criteria; and, scheduling sampling among a plurality of threads via software.

14. (Currently Amended) A method of sampling instructions executing in a multi-threaded processor comprising:

setting a candidate counter to a number;

selecting an instruction for sampling;

storing information relating to the instruction;

determining whether all events for the instruction have occurred;

determining whether the instruction includes events of interest, the events of interest including whether the instruction corresponds to a desired sampled thread;

decrementing the candidate counter when all events for the instruction have occurred and when the instruction includes events of interest;

determining whether the candidate counter equals zero; and

reporting the instruction to a particular thread when the candidate counter equals zero.

15. (Original) The method of claim 14 further comprising providing a register with a bit vector representing events of interest; and

wherein the determining whether the instruction includes events of interest further includes comparing the information relating to the instruction to the bit vector.

16. (Original) The method of claim 14 wherein the information relating to the instruction represents an instruction history, and the instruction history includes information relating to at least one of an event value, a program counter value, a branch target address value, an effective memory address value, a latency value, a number in issue bundle value, a number in retire bundle value, a privileged value, a branch history value and a number in fetch bundle value.

17. (Original) The method of claim 14 wherein the selecting an instruction for sampling is based upon sample selection criteria; and

the sample selection criteria include information relating to a desired sampled thread.

18. (Currently Amended) A multi-threaded processor comprising:

a sampling logic configured to determine whether an instruction executed in the processor corresponds to a desired sampled thread;

a sampling register logic coupled to the sampling logic;

an instruction history register logic coupled to the sampling register logic, the instruction history register logic storing information relating to the instruction;

a sample filtering and counting logic coupled to the sampling logic,[[; and]]

wherein the sample filtering and counting logic is replicated on a per thread basis[[.]]; and

a notification logic, the notification logic reporting to a particular thread the information relating to the instruction if the instruction corresponds to the desired sampled thread.

19. (Cancelled).

20. (Previously Presented) The processor of claim 18 wherein the sampling register logic includes a register with a bit vector representing events of interest; and

wherein the sampling logic determines whether the instruction includes events of interest by comparing the information relating to the instruction to the bit vector.

21. (Previously Presented) The processor of claim 18 wherein the information relating to the instruction represents an instruction history, and the instruction history includes information relating to at least one of an events value, a program counter value, a branch target address value, an effective memory address value, a latency value, a number in issue bundle value, a number in retire bundle value, a privileged value, a branch history value and a number in fetch bundle value.

22. (Previously Presented) The processor of claim 18 wherein the sampling register logic includes a sample selection criteria register storing sample selection criteria; and the sample selection criteria include information relating to a desired sampled thread.

23. (New) The method of claim 1 wherein storing the sampling information further comprises storing the sampling information to a memory register shared by multiple threads.